

IN THE CLAIMS:

Please amend the claims as follows:

1-9. (Canceled)

10. (Currently Amended) A method of providing a user access to functional modules from within a[[n]] query application, ~~used to build queries, issue queries and/or, view query results, during a query session~~ comprising:

assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database, wherein the assigned metadata requirements specify conditions required for successful execution of the functional module, wherein at least one condition defines at least one user role required for successful execution of the functional module;

collecting runtime metadata relating to one or more result fields ~~[[of]]~~in a query statement, wherein the one or more result fields specify one or more data fields for which data is requested to be returned upon execution of the query statement, wherein the runtime metadata is collected after composition of the query statement, and wherein the runtime metadata is collected before the query statement is submitted for execution;

obtaining a list of functional modules that are accessible from within an application used during the query session;

identifying a limited subset of the functional modules in the list that will successfully execute, by comparing the collected runtime metadata with the assigned metadata requirements; and

providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute.

11-13. (Canceled)

14. (Previously Presented) The method of claim 10, wherein obtaining metadata associated with the functional module comprises examining a signature validation.

15. (Previously Presented) The method of claim 10, wherein the metadata associated with at least one of the functional modules comprises at least one of: one or more input parameters required for successful execution of the functional module, one or more output parameters required for successful execution of the functional module, and a security credential required to execute the functional module.

16. (Original) The method of claim 10, wherein at least one of the functional modules analyzes query results.

17-19. (Canceled)

20. (Currently Amended) A computer readable storage medium containing a program which, when executed, performs operations for providing a user access to functional modules from within an application, comprising:

assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database, wherein the assigned metadata requirements specify conditions required for successful execution of the functional module, wherein at least one condition defines at least one user role required for successful execution of the functional module;

collecting runtime metadata relating to one or more result fields [[of]]in a query statement, wherein the one or more result fields specify one or more data fields for which data is requested to be returned upon execution of the query statement, wherein the runtime metadata is collected after composition of the query statement, and wherein the runtime metadata is collected before the query statement is submitted for execution;

obtaining a list of functional modules accessible from within the application;

identifying a limited subset of functional modules that will successfully execute, by comparing the collected runtime metadata with the assigned metadata requirements; and

providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute.

21. (Previously Presented) The computer readable storage medium of claim 20 wherein the application is a query building application.

22-26. (Canceled)

27. (Currently Amended) A data processing system for providing a user access to functional modules from within an application comprising:

a data repository;

a plurality of functional modules, each having associated metadata requirements that specify conditions required for successful execution of the functional modules, wherein at least one condition defines at least one user role required for successful execution of the functional modules;

an application from which the functional modules are accessible, wherein the application is configured to:

after composition of a query statement, but before the query statement is submitted for execution, collect runtime metadata related to one or more result fields [[of]]in the [[a]]query statement, wherein the one or more result fields specify one or more data fields for which data is requested to be returned upon execution of the query statement~~after composition of the query;~~ and

present to a user a limited subset of the functional modules that will successfully execute, as determined by the application based on the collected runtime metadata and the metadata requirements associated with the functional modules.

28. (Original) The data processing system of claim 27, wherein the data repository comprises XML data structures used to store runtime metadata.

29. (Original) The data processing system of claim 27, wherein the data repository comprises relational database tables used to store runtime metadata.

30. (Currently Amended) The method of claim 10, wherein the runtime metadata relating to one or more result fields in ~~[[of]]~~ the query statement comprises one or more of:

a result field name; and

a data type for the result field; ~~and~~

~~— a number of records returned.~~

31. (Currently Amended) The computer readable storage medium of claim 20, wherein the runtime metadata relating to one or more result fields in ~~[[of]]~~ the query statement comprises one or more of:

a result field name; and

a data type for the result field; ~~and~~

~~— a number of records returned.~~

32. (Currently Amended) The data processing system of claim 27, wherein the runtime metadata related to one or more result fields in ~~[[of]]~~ the query statement comprises one or more of:

a result field name; and

a data type for the result field; ~~and~~

~~— a number of records returned.~~